

Automated Planning and Scheduling for Goal-Based Autonomous Spacecraft

Steve Chien, Benjamin Smith, and Gregg Rabideau, Jet Propulsion Laboratory
Nicola Muscettola and Kanna Rajan, NASA Ames Research Center

1094-7167/98/\$10.00 ©1998 IEEE

AUTOMATED PLANNING AND SCHEDULING TECHNOLOGY ENABLES A NEW CLASS OF AUTONOMOUS SPACECRAFT. THE AUTHORS DISCUSS THE BENEFITS OF THIS TECHNOLOGY AND OFFER AN OVERVIEW OF ITS USE AT NASA.

Automated planning and scheduling technology—we'll call it *automated planning systems*, for the sake of brevity—is applicable to a wide spectrum of spaceflight missions, from those with limited onboard computational capabilities, such as Lunar Prospector, to those with highly sophisticated software, such as Cassini. In all cases, the goal is for the mission scientist to command the spacecraft directly, with no need for mission operations specialists to perform routine activities.

Routine use of automated planning systems for both ground and onboard operations can greatly benefit spacecraft missions in several ways.

- *Reduced costs.* Automated planning systems all but eliminate the need for the mission operation's team to manually generate command sequences, dramatically reducing costs. According to one estimate, automated commands can reduce the cost of mission operations by as much as 60% (excluding data analysis).¹ Our recent experiences support these projections. For example, using the Dcaps automated planning system to command the Data-Chaser shuttle payload reduced commanding-related mission operations effort by 80%² compared to manual sequence generation.
- *Increased responsiveness.* With planning-systems technology, a goal-based spacecraft can more readily perform opportunistic science. When an unexpected opportunity occurs (such as a supernova or solar phenomena) the spacecraft can immediately perform appropriate measurements rather than wait until the ground operations team detects the event and uplinks commands to the spacecraft.
- *Increased interactivity.* A goal-based autonomous spacecraft can facilitate interactive science. A self-commanding spacecraft can perform high-level science requests, such as "Perform an interferometry sweep with priority 5." A direct connection and faster feedback between scientist and spacecraft create a new model for scientific discovery in space.
- *Increased productivity.* Automated-planning-systems technology has the potential to increase science return. It does this by producing operations plans that better optimize the use of scarce science resources. For example, the Dcaps planning system increased science return by 40% over manually generated sequences.²
- *Simplified self-monitoring.* Planning-systems technology simplifies self-monitoring, onboard fault-management, and spacecraft-health tasks. Because the spacecraft can respond directly without waiting for ground communication, it can cover a greater range of faults.

In this article, we describe our use of symbolic AI in planning systems, provide an overview of the spacecraft-operations domain, and discuss several past, ongoing, and future deployments of planning-systems technology at NASA.

Planning systems and symbolic AI

We use symbolic AI techniques to solve space-operations planning and scheduling problems because its declarative representations allow for explanation and introspection of produced plans.

Design constraints. Using symbolic AI commits us to two design constraints. The first is that we must use a declarative model to represent spacecraft and mission constraints, including explicit models of spacecraft subsystem modes and resource capacity. Second, symbolic AI performs search to find solutions. In the spacecraft-operations domain, solutions are defined as a projected evolution of actions and states that begins in the current state and satisfies goals posed to the system.

Given these constraints, the planning system works as follows. The planning system addresses one flaw at a time from its list of current flaws in the plan, such as unachieved top-level goals proposed to the planning system, unachieved conditions required by desired states or activities in the plan, or, in some cases, oversubscribed resources. It begins by categorizing the flaw type, then changes the plan until the flaw is resolved. Next, it updates the plan database, which reflects the plan's projected evolution of states and resources. The planning system then recomputes the current flaw list. This process repeats until the planning system produces a flawless plan.

Some search methods, such as refinement search, can generate a dead-end in which no existing, legal changes will resolve a flaw. The planning system must then backtrack and alter a previous choice. Other search methods, such as iterative repair, achieve a similar effect by applying a step that undoes a previous decision.

Language expressiveness. Given the existing design constraints, the expressiveness of the language used to represent the spacecraft, the mission constraints, and the plan is a critical choice. For example, some representations allow complex functional dependencies between state variables or resources, while others do not. Some representations permit partial constraints on state or activity parameters or timings, while others fix activity and state timings to precise values.

An expressive representation is advantageous because it describes complex behaviors compactly. For example, a plan might use an explicit representation of timing constraints between events, from which it can deduce flexible time windows for each event. The plan-execution system can exploit this intrinsic flexibility to tolerate timing fluctuations and some unexpected events.

Expressive representation also allows problem solving at more abstract levels, with each decision corresponding to a larger set of spacecraft behaviors. More abstract problem solving means that fewer decisions must be made; this can yield a large (possibly exponential) increase in speed.

However, the added power of expressiveness can be costly. As a language's expressive power increases, so does the cost of determining a plan's correctness. In general, to determine whether a plan will execute correctly, values must be assigned consistently to all plan variables. When the plan includes more and more general classes of constraints, finding such a value assignment can be computationally expensive. Although researchers have done considerable work in assessing the impact of plan representation on computational efficiency, their work has focused on classical

plan representations, such as no metric time, aggregate resources, or functional dependencies among plan variables. Rao Kambhampati's recent paper provides a starting point for investigating this body of work.³

The best balance between expressiveness and problem-solver performance varies according to the application; finding the most effective balance remains an area of active research.

Within our planning-systems framework, we have not yet addressed issues of representing and reasoning about plan quality and optimization. As with most domains, planning systems for spacecraft operations have both hard constraints, which a plan must satisfy, and soft constraints or preferences, which a planning system must optimize. We are investigating how to represent, reason about, and optimize these soft constraints.

The spacecraft-operations domain

Spacecraft mission operations presents a number of unique challenges and opportunities for planning systems.

Knowledge integration. In our domain, the planning system must integrate knowledge from a wide range of sources. Because many spacecraft-operations constraints involve specialized reasoning that is difficult to represent in planning languages, the planning system must often integrate external constraints and knowledge (see the sidebar, "Further reading" for more on this). For example, many spacecraft activities require that specific spacecraft parts point at specific objects, such as when a science observation requires an instrument to point at an asteroid or when downlinking data requires the high-gain antenna to point at earth. To determine which pointing requirements are possible, when they are possible, and how much time it takes to turn requires significant amounts of geometric reasoning. Thus, the planning system must be able to communicate with special reasoning algorithms and incorporate the results.

//insert sidebar, "Further reading," here//

Resource constraints. Building spacecraft capable of flying hundreds of millions of miles in extreme environmental conditions is extremely difficult and expensive. For example, to generate more power onboard, a spacecraft requires either larger solar panels (to create more solar power) or a larger nuclear power unit (to create more nuclear power). Both of these options significantly increase the spacecraft's weight and volume, dramatically increasing the mission's cost. As a result of such situations, spacecraft resources are scarce and in high demand. Automated planning systems can lead to more efficient use of spacecraft resources to achieve science goals, and it can do so for less than the cost of improving spacecraft design. Clearly, it can add considerable value to a mission.

Reliability. Space-exploration missions can cost upwards of several hundred million dollars. Because an incorrect command sequence could damage or destroy the spacecraft, reliability is crucial. To avoid failure, planning-systems development requires formal verification methods for the problem solver and planning model, as well as extensive and rigorous testing of the planning system under most plan-generation conditions.

Multiphase use. A planning system is valuable in multiple phases of a mission lifecycle. In preliminary mission planning, which begins years before a launch, systems engineers can use the planning system to approximate models of spacecraft resource use and science return to assess different hardware configurations, launch trajectories, and science-gathering strategies. As launch approaches, the planning system can help refine the spacecraft, operations, and science models,

and update plans accordingly. After launch, during the actual primary-mission lifetime, the planning system can generate spacecraft-operations plans. If a mission is extended, the planning system can help minimize the operations staff required to perform continuing science goals.

NASA's planning systems projects

Our work has been part of a larger effort at NASA to extend and deploy automated-planning technology for spacecraft command. Following is an overview of three such systems.

Dcaps. Data-Chaser is a shuttle payload that flew onboard Space Shuttle flight STS-85 in August 1997. This payload carried several science instruments and demonstrated distributed science-driven commanding.

Data-Chaser included ground-based automated-planning-systems technology. The Data-Chaser Automated Planning System (Dcaps) was developed jointly by JPL and the University of Colorado (which built and operated the Data-Chaser payload). Dcaps used the Plan-It2 sequencing tool,⁴ which provides an extensive modeling capability. Plan-It2 uses a state and plan representation that is totally committed on activity times and parameter values. Dcaps added a domain-specific, initial-schedule-generation capability and a general iterative-repair capability to Plan-It2.

The Dcaps model of the Data-Chaser payload was fairly large and included 67 resources and states and 58 activity types. Dcaps-generated commands were Spacecraft Command Language scripts, which were used to implement the Data-Chaser execution system. Dcaps commands ranged from fairly high-level commands, such as performing an observation (including opening the instrument door, reading the instrument, closing the door, and transferring the data) to low-level commands such as tripping a relay.

The ground team had tremendous success using Dcaps to automatically generate commands for the Data-Chaser payload. For the first five days of the STS-85 flight, the Data-Chaser operations team manually generated command sequences. For the last seven days of the flight, they used the Dcaps automated capability. Compared to the manual approach, Dcaps reduced by 80% the operations-team labor required to generate a six-hour operations plan. Also, it increased science return by 40% per six-hour operations window. Thus, Dcaps demonstrated that planning-systems technology is mature enough to significantly improve operations. It also showed that basic planning technologies—such as initial-schedule generation and basic heuristic iterative repair—can be of considerable value in operations.

The PS planning system. NASA's New Millennium Program is an aggressive series of missions intended to demonstrate breakthrough technologies for space exploration. The program consists of two tracks. The first, led by JPL, is targeted at deep-space exploration; the other, led by the Goddard Space Flight Center, is aimed at earth orbiters.

The NMP's first mission, New Millennium Deep Space One (NM DS1), is scheduled to launch in October 1998 and will fly by an asteroid, Mars, and a comet while demonstrating several new technologies, including solar electric propulsion. The NMP will also demonstrate the Remote Agent technology,⁵ the first prototype of a complete control system for an autonomous spacecraft based on AI technologies.

The Remote Agent consists of three distinct modules:

- a mode-identification and recovery system, which detects and corrects execution-time anomalies (such as valves or switches stuck open or close);

- an intelligent task executive, which coordinates Remote Agent operations (including real-time control software) and can execute flexible plans; and
- the Planner Scheduler planning system (described below).

The Remote Agent will assume experimental control of the spacecraft for one week, three to four months after launch. The Remote Agent's planning system, called Planner Scheduler (PS),⁶ is an evolution of the Heuristic Scheduling Testbed System (HSTS)⁷ planning system. The HSTS planning system was developed jointly by JPL and NASA Ames Research Center, which led the project. PS consists of the HSTS temporal database and modeling facility, developed at NASA ARC, and the Incremental Refinement Search Engine (IRS), a backtracking, refinement search engine developed primarily by JPL.

PS allows for expression of complex temporal constraints⁷ and contains a powerful suite of constraint-propagation capabilities. PS supports complex functional dependencies and has powerful facilities for partially grounding parameters. The HSTS plan representation maintains least-constraining temporal intervalism as in a Simple Temporal Problem.⁸

The IRS search engine implements refinement search. In this approach, each search node is more strictly constrained than its parent node. IRS can be programmed for efficiency using an expressive language for specifying heuristics⁹ at the various choice points in the search.

HSTS favors least-commitment search and constraint propagation, which has two potential advantages. First, if properly managed, a least-commitment approach can considerably reduce the search (for more on this, see the "Further reading" sidebar). Second, it generates intrinsically flexible plans; during the search process, a least-commitment planning system attempts to introduce constraints only when it is unavoidable. A smart executive will flexibly execute the plan to react to timing fluctuations in an event.

In the Remote Agent, PS limits execution-time flexibility to the timing of a start or end event in an activity or state. PS also pays attention to how rapidly the smart executive executes a plan. To improve the executive's real-time responsiveness, PS post-processes the plan's temporal constraint network so that the smart executive can process the network as quickly as possible.^{10,11}

For modeling-support, PS relies on the HSTS modeling language, which favors a first-principles approach to planning and has a precise, elegant semantics.¹² In relying on HSTS, PS relies purely on generative planning and does not use task reduction or hierarchical task network (HTN) planning. This choice is part of the HSTS design philosophy, which aims to cleanly distinguish between allowable transitions and heuristics. HTN encoding can be considered a form of search control. Although researchers have formalized HTN planning approaches, these efforts have focused on classical planning models, and thus are far from the more "industrial strength" HTN planning systems such as SIPE¹³ or O-Plan.¹⁴ For more information on these and other issues in modeling support, see the "Further reading" sidebar.

The distinction between allowable transitions and heuristics is important because it facilitates domain-knowledge validation. In the PS approach, systems engineers and mission specialists need only inspect the domain-knowledge constraints, leaving the task of designing the heuristics to optimize performance to the planning-algorithm specialist.

An enhanced version of the DSI Remote Agent with the PS planning system is scheduled to serve as the primary control system for the NM DS3 mission, launching in 2001. The mission will consist of three formation-flying spacecraft that will perform interferometric measurements. Ground operations will control the NM DS3 spacecraft by uplinking a set of goals to the Remote Agent.

Aspen. The Aspen¹⁵ planning system is being demonstrated in several contexts. The UHF Follow-on One (UFO-1) spacecraft is an in-orbit test bed for spacecraft autonomy operated by the United States Naval Academy Space Artificial Intelligence Laboratory (USNA SAIL). In a collaborative effort between JPL and USNA SAIL, researchers are using Aspen to increase the automation levels in ground-based command generation. The first Aspen demonstration is scheduled for fall 1998; a series of demonstrations will culminate in fully automated, lights-out commanding (in which there will be no ground-operations personnel for routine commanding) in early 1999.

Aspen represents an evolutionary approach closer to current spacecraft commanding and sequencing paradigms. Aspen uses an action-centered representation language¹⁶ designed by non-AI personnel. In this language, state and resource changes are directly attributed to actions and events. Aspen's modeling language lets systems engineers encode standard operating procedures as task decompositions. However, as a result of this more user-oriented approach, Aspen lacks clean, precise semantics. Like HSTS, Aspen lets temporal constraints be expressed as a Simple Temporal Problem (STP).⁸ It also allows for expression of complex parameter constraints. Unlike HSTS, Aspen does not support partial grounding; it prefers a more committed state representation, requiring grounding parameter values and action timings to track state and resource use.

Although Aspen supports certain elements of constraint propagation and least-commitment search, its bias is toward committed search, for several reasons:

First, committed search places fewer requirements on the structure and complexity of auxiliary, special-purpose reasoning modules. In spacecraft commanding, these modules provide the planning system with information about spacecraft functions such as navigation, attitude control, power management, and thermal-constraint management. The simplest way for these modules to operate is to return a specific value for each query.

For example, the planning system might need to ask the attitude-control module for a turn path that will safely turn the spacecraft from one pointing direction to another. The exact turn path can vary significantly, depending on its exact start time, because certain spacecraft areas must not turn toward certain celestial bodies (the camera must not face the sun, for example). This is further complicated in that the relative position of celestial bodies varies over time, particularly during a flyby. To fully exploit the power of a partially constrained plan representation, the planning system must use a more abstract information description: instead of a single turn path, the planning system needs a range of start times and path durations that stay within prespecified limits. However, building such abstractions can be costly and might not give the approximation level necessary to guarantee planning-system consistency in every possible execution. The balance between additional abstraction effort and potential payoff in terms of reduced search and solution flexibility will vary, depending upon the domain.

Second, spacecraft often have numerous aggregate resources; it is computationally difficult to reason about aggregate resources in an uncommitted representation. Third, committed schedules are more easily verified using conventional spacecraft-sequencing tools (discrete-event simulations). These tools simulate the spacecraft's resource and state evolution with higher-fidelity models to verify that an uplinked sequence will achieve the desired objectives without harming the spacecraft. Fourth, committed schedules facilitate visualization for mixed-initiative planning, which is useful in ground automation.

Finally, in some cases, a more committed representation allows faster search, including less time per search node visited. This is consistent with recent use of propositional satisfiability (SAT) representations for planning.¹⁷ The reduced cost per search node must be balanced against

more informed or stronger constraint-propagation approaches that can reduce the number of search modes required, but that often have a high cost per search node visited.

Aspen is scheduled to be used to automate ground operations for GSFC's New Millennium Earth Orbiter One (NM EO-1) spacecraft, launching in 1999, as well as for NM EO-1's extended mission, which begins in 2000. Figure 1 shows an Aspen screen shot of a one-week operations plan for the NM EO-1 spacecraft. The Aspen display is time-oriented; later times are displayed to the right on the horizontal axis. The upper portion of the screen shows the activities in the current mission plan, with each line beginning at the activity's start time and ending at its end time. The timelines toward the bottom of the display show the state and resource evolution as modeled and tracked by the Aspen planning software.

//Insert c3chi01.eps here//

Figure 1. A one-week operations plan for the New Millennium Earth Orbiter One generated by the Aspen planning system.

As these experiences show, AI has clearly had an impact on the spacecraft-operations community, particularly in the model-based, declarative approach of AI planning and how it facilitates model construction, validation, and refinement as compared to current procedural methods such as scripting.

Still, several questions with deep implications for AI remain open. One fruitful area of exploration concerns finding the right balance between a plan's flexibility and robustness and the speed of the computational substrate needed to construct and execute the plan.

Another area of current work is how to integrate planning and execution to allow for responsiveness to run-time variations. One approach is to build such flexibility into the plan, verifying before execution that such flexibility will respect the plan's causal structure. Another approach is to retain the general causal structure, but to ground times in the plan while verifying it, and to replan and revalidate (using the general causal structure) when runtime variations occur.

Yet another area of current work involves representing and reasoning about soft constraints in space operations plans. Representing preferences and generating optimizing planners is an exciting area of ongoing work.

A final area of work is aimed at gaining a better understanding of the relationship between spacecraft operations planning and other types of scheduling (such as manufacturing). For example, spacecraft operations differs from other scheduling domains, such as production management. In production-flow scheduling, the objective is often to complete a set of work as quickly as possible. Although the domains share some common elements (such as problem-representation or constraint-propagation methods), in spacecraft operations the opportunities are less contiguous. Thus, while the operations team might attempt to make as many observations as possible in any single opportunity, the set of opportunities is a set of discrete, disjoint science opportunities dictated by geometries. Because of this difference, we expect the techniques and heuristics in spacecraft operations will vary more in this domain than in manufacturing planning.

Acknowledgement

Portions of this research were carried out by the Jet Propulsion Laboratory under a contract with the National Aeronautics and Space Administration.

References

1. R. Ridenoure, "New Millennium Mission Operations Study," JPL Internal Document, June 1995; available at <<forthcoming from author.>>
2. S. Chien et al., "Automating Planning and Scheduling of Shuttle Payload Operations," submitted to the *Artificial Intelligence Journal* Special Issue on Applications; available at <http://www-aig.jpl.nasa.gov>.
3. R. Kambhampati, "Challenges in Bridging Plan Synthesis Paradigms," *Proc. IJCAI '97: Int'l Joint Conf. AI*, Morgan Kaufmann, San Francisco, 1997, pp. 444-49.
4. W.C. Eggemeyer, "Plan-It-II Bible," Internal JPL Document; available at <<forthcoming from author.>>
5. B. Pell et al., "An Autonomous Spacecraft Agent Prototype," *Autonomous Robots*, Vol. 5, No. 1, Mar. 1998.
6. N. Muscettola et al., "On-Board Planning for the New Millennium Deep Space One Spacecraft," *Proc. 1997 IEEE Aerospace Conf.*, IEEE Computer Society Press, Los Alamitos, Calif., 1997, pp. 303-318.
7. N. Muscettola, "HSTS: Integrating Planning and Scheduling," *Intelligent Scheduling*, M. Fox and M. Zweben, eds., Morgan Kaufmann, San Francisco, 1994, pp. 169-212.
8. R. Dechter, I. Meiri, and J. Pearl, "Temporal Constraint Networks," *Artificial Intelligence*, No. 49, 1991, pp. 61-95.
9. B. Smith, K. Rajan, and N. Muscettola, "Knowledge Acquisition for the Onboard Planner of an Autonomous Spacecraft," *Lecture Notes in Artificial Intelligence*, Springer-Verlag, Berlin, 1997.
10. N. Muscettola, P. Morris, and I. Tsamardinos, "Reformulating Temporal Plans for Efficient Execution," *Proc. KR '98: Sixth Int'l Conf. Principles of Knowledge Representation and Reasoning*, AAAI Press, Menlo Park, Calif., 1998.
11. I. Tsamardinos, N. Muscettola, and P. Morris, "Fast Transformation of Temporal Plans for Efficient Execution," *Proc. AAAI '98*, AAAI Press, New York, 1998, pp. 254-259.
12. A. Cesta and A. Oddi, "DDL.1: A Formal Description of a Constraint Representation Language for Physical Domains," *New Directions in AI Planning*, M. Ghallab and A. Milani, eds., IOS Press, Amsterdam, 1996.
13. D. Wilkins, *Practical Planning: Extending the Classical AI Planning Paradigm*, Morgan Kaufmann, San Francisco, 1988.
14. A. Tate, B. Drabble, and R. Kirby, "O-Plan2: An Open Architecture for Command Planning and Control," *Intelligent Scheduling*, M. Fox and M. Zweben, eds., Morgan Kaufmann, San Francisco, 1994, pp. 213-240.
15. A. Fukunaga et al., "Towards an Application Framework for Automated Planning and Scheduling," *Proc. Of the IEEE Aerospace Conference*, IEEE Computer Society Press, Los Alamitos, Calif., 1997, p. 375-386.
16. R. Sherwood et al., "Using Aspen to Automate EO-1 Activity Planning," *Proc. 1998 IEEE Aerospace Conf.*, IEEE Press, Piscataway, N.J., 1998; available at <http://eazy.jpl.nasa.gov/sherwood/pp>.
17. H. Kautz and B. Selman, "Pushing the Envelope: Planning, Propositional Logic, and Stochastic Search," *Proc. AAAI '96*, AAAI Press, Menlo Park, Calif., 1996, pp. 1194-1199.

Steve Chien is a principal computer scientist and technical group supervisor of the Artificial Intelligence Group in the Information and Computing Technologies Research Section of the Jet Propulsion Laboratory, California Institute of Technology. He leads several projects in research and development of advanced planning systems for spacecraft operations. His research interests are in the areas of planning and scheduling, machine learning, operations research, and decision theory. Chien received a PhD in Computer Science from the University of Illinois. Contact him at Steve.chien@jpl.nasa.gov

Nicola Muscettola is a senior computer scientist at the Computational Sciences Division of the NASA Ames Research Center. He is the principal designer of the HSTS planning framework and is the lead of the on-board planner team for the Deep Space 1 Remote Agent Experiment. His research interests include planning, scheduling, temporal reasoning, constraint propagation, action representations and knowledge compilation. Muscettola received his Diploma di Laurea in electrical and control engineering and his PhD in computer science from the Politecnico di Milano, Italy. Contact him at Mus@ptolemy.arc.nasa.gov

Kanna Rajan is a member of the Planning and Scheduling Group at NASA Ames Research Center. Prior to joining NASA Ames, he was in the doctoral program in computer science at the Courant Institute of Mathematical Sciences at New York University. His primary research interests are in planning, robotics and knowledge representation. Rajan received a BS from the Birla Institute of Technology and Science, Pilani, India, and an MS from the University of Texas, Arlington, both in computer science. Contact him at Kanna@ptolemy.arc.nasa.gov

Ben Smith is a member of the technical staff in the Artificial Intelligence Group at the Jet Propulsion Laboratory. He is lead of the JPL element of the DS1 planning team. His research interests include intelligent agents, machine learning,

and planning. Smith holds a PhD in computer science from the University of Southern California. Contact him at Benjamin.smith@jpl.nasa.gov

Gregg Rabideau is a member of the technical staff in the Artificial Intelligence Group at the Jet Propulsion Laboratory. His research interests are in the areas of planning and scheduling for spacecraft commanding. Gregg received a BS and an MS in computer science from the University of Illinois. Contact him at Gregg.rabideau@jpl.nasa.gov

Further Reading

There is a wealth of research available on automated planning systems; here we provide pointers to further reading on specific topics related to our approach.

Knowledge integration

- B. Pell et al., "An Autonomous Spacecraft Agent Prototype," *Autonomous Robots*, Vol. 5, No. 1, Mar. 1998.
- N. Muscettola et al., "On-Board Planning for the New Millennium Deep Space One Spacecraft," *Proc. 1997 IEEE Aerospace Conf.*, IEEE Computer Society Press, Los Alamitos, Calif., 1997, pp. 303-318.
- R. Sherwood et al., "Using Aspen to Automate EO-1 Activity Planning," *Proc. 1998 IEEE Aerospace Conf.*, IEEE Press, Piscataway, N.J., 1998; available at <http://eazy.jpl.nasa.gov/sherwood/pp>.

Least-commitment search approach

- A. Barrett and D. Weld, "Partial Order Planning: Evaluating Possible Efficiency Gains," *Artificial Intelligence* 67, May 1994, pp. 71-112.
- S. Smith and C. Cheng, "Slack-Based Heuristics for Constraint Satisfaction Scheduling," *Proc. AAAI '93*, AAAI Press, New York, 1993, pp. 139-144.
- S. Smith and C. Cheng, "A Constraint Satisfaction Approach to Makespan Scheduling," *Proc. AIPS '96: Int'l Conf. AI Planning Systems*, AAAI Press, Menlo Park, Calif., 1996, pp. 45-52.
- N. Muscettola, "HSTS: Integrating Planning and Scheduling," *Intelligent Scheduling*, M. Fox and M. Zweben, eds., Morgan Kaufmann, San Francisco, 1994, pp. 169-212.
- A. Baker, "The Hazards of Fancy Backtracking," *Proc. AAAI '94*, AAAI Press, New York, 1994, pp. 288-293.

HSTS modeling support

- K. Erol, J. Hendler, and D. Nau, "UMCP: A Sound and Complete Procedure for Hierarchical Task-Network Planning," *Proc. AIPS'94: Int'l Conf. AI Planning Systems*, AAAI Press, Menlo Park, Calif., 1994, pp. 249-255.
- A. Lansky, "Action-Based Planning," *Proc. AIPS'94: Int'l Conf. AI Planning Systems*, AAAI Press, Menlo Park, Calif., 1994, pp. 110-115.
- Barrett, *Task-Decomposition Planning with Context-Sensitive Actions*, PhD dissertation, Dept. of Computer Science, Univ. of Washington, Seattle, Wash., 1997.